REMARKS

This AMENDMENT is filed in reply to the outstanding Office Action of October 29, 2003, and is believed to be fully responsive thereto for reasons set forth below in greater detail.

In the Office Action, the Examiner rejected Claims 1-3,6-8, 10, 12, 25-27, 30, 32 and 35 under 35 U.S.C. §103(a), as allegedly being unpatentable over Dean et al. (U.S. Patent No. 6,374,367)(hereinafter "Dean1") in view of Dean et al. (U.S. Patent No. 6,070,009) (hereinafter "Dean2"). The Examiner additionally rejected Claims 9 and 31 under 35 U.S.C. §103(a), as allegedly being unpatentable over Dean1 in view of Dean2 and further in view of Bala (U.S. Patent No. 6,351,844) (hereinafter "Bala"). The Examiner additionally rejected Claims 4-5 and 28-29 under 35 U.S.C. §103(a), as allegedly being unpatentable over Dean1 in view of Dean2 and further in view of Brown (U.S. Patent No. 4,853,884) (hereinafter "Brown"). The Examiner additionally rejected Claims 11 and 33 under 35 U.S.C. §103(a), as allegedly being unpatentable over Dean1 in view of Dean2 and further in view of McDevitt et al. (U.S. Patent No. 6,266,678). The Examiner further rejected Claims 13, 15-17, 20-21, 24, 34, and 36 under 35 U.S.C. §103(a), as allegedly being unpatentable over Dean1 in view of Dean2 and further in view of Holzle et al. (U.S. Patent No. 5,995,754) (hereinafter "Holzle"). The Examiner additionally rejected Claims 18-19 under 35 U.S.C. §103(a), as allegedly being unpatentable over Dean1 in view of Dean2 and further in view of Holzle and Brown. The Examiner additionally rejected Claim 14 under 35 U.S.C. §103(a), as allegedly being unpatentable over Dean1 in view of Dean2 and further in view of the reference to Alpern et al. entitled "The Jalapeno Virtual Machine", IBM Systems Journal, Vol. 39, No. 1, February 2000 (hereinafter "Alpern"). Finally, the Examiner rejected Claim 22 under 35

U.S.C. §103(a), as allegedly being unpatentable over Dean1 in view of Dean2 and further in view of Holzle and further in view of McDevitt and further rejected Claim 23 as allegedly being unpatentable over Dean1 in view of Dean2 and further in view of Holzle and further in view of Alpern.

As a preliminary matter, Applicants take this opportunity to correct minor informalities in the specification, for instance, at pages 9 and 11, by entering current U.S. Patent Application Serial Numbers for referenced commonly-owned, co-pending patent applications.

With respect to the substantive rejections of independent Claims 1, 15 and 25 under 35 U.S.C. §103(a), the Applicants' respectfully traverse.

The thrust of the applicants' traversal is that both Dean1 and Dean2 teach away from the approach for characterizing program behavior as the approach taken by applicants' invention. Particularly, both Dean1 and Dean2 teach in the background section of each of their respective disclosures (e.g., column 2, lines 4-9 of Dean1) that instrumentation (i.e., inserting of additional code (e.g., yield points) to executing programs) has drawbacks. Dean1 and Dean2 in their respective embodiments thus, disqualify this approach and teach a hardware-based sampling approach for obtaining profile data. Given these teachings, one of ordinary skill in the art would not think of using yield points (additional code) to obtain profile data.

The present invention, on the other hand, as set forth in Claims 1, 15 and 25 set forth a purely software-based approach for obtaining profile data. To clarify a salient feature of the invention, Claims 1, 15 and 25 have been amended to set forth a mechanism for inserting yield points at distinguished locations of a program to be executed, each yield point indicating a potential sampling operation during execution of said program. Such mechanism

may include, for example, a compiler, however other means, such as manually, are also comtemplated. The addition of the method step of inserting yield points at distinguished program points highlights the software-based approach of the invention which is patentably distinct from the hardware only approach taught by Dean1 and Dean2.

Applicants respectfully assert that no new matter is being entered by the amendments to independent Claims 1, 15 and 25 and full support for these amendments is provided in the specification at page 5, lines 4-7 and page 6, lines 5-16.

Thus, as amended independent Claims 1 and 25 now set forth novel features of a complete operative software-based method and system for characterizing runtime behavior of an executing computer program having elements that are not taught by Dean1 and Dean2, the Examiner is respectfully requested to withdraw the rejection of Claims1 and 25 under 35 U.S.C. §103(a) and all claims dependent therefrom.

With respect to the rejection of independent method Claim 15, the Examiner had further applied the Holzle reference. It is respectfully submitted that Holzle implements a counter-based approach to determine when to take an action: that is, "how many times a specific portion of compiled code has been called". In this approach, Holzle uses one counter per method. In the software-based approach of the invention as claimed in amended Claim 15 a counter device is implemented for counting a number of identified yield points which is then compared against a predetermined threshold. However, the counting approach of the invention only implements one counter. In addition, when one of Holzle's counters reaches a threshold, Holzle teaches recompiling the associated method. In contrast (as per Claim 15, amended element e)), when the counter implemented in applicants' invention reaches a threshold, applicants' system performs a sampling operation of the executing program (i.e., runtime state), and, records relevant information for characterizing behavior of the execution

environment in response to the sampling. Moreover, Holzle is not concerned with distinguished program points, but whether or not a method is invoked. To the contrary, in the approach of the present invention, yield points may be placed anywhere within a method and the counter can be incremented multiple times for a method invocation.

In view of the foregoing, the Examiner is respectfully requested to withdraw the rejection of independent Claim 15 under 35 U.S.C. §103(a) and all claims dependent therefrom.

With respect to the Examiner's rejection of Claim 9, the Examiner alleges that Bala's timer interrupt and interrupt handling scheme is related to the approach as set forth in Claims 8 and 9. Respectfully, in the present invention, an executing program is only interrupted at distinguished program points. In a described embodiment of the present invention, when a timer goes off, a bit is set in the system state. When a yield point is encountered (executed), the bit is checked. Only if the bit is set is any action taken to sample the runtime data. As in Dean1 and Dean2, Bala does not use instrumentation to identify distinguished program points where the program can be interrupted, but interrupts a program at any program point when a timer goes off (See Bala at column 5, lines 61-67). In addition, like Dean1 and Dean2, Bala asserts that instrumentation (software insertion) is not a desired implementation due to code explosion and overhead (See Bala at column 5, lines 26-31). Like Dean1 and Dean2, Bala thus teaches away from the present invention.

Thus, further, with respect to the rejection of Claim 7, the trigger bit that is set in the present invention to indicate whether a sampling operation to ascertain software state is to take place, is not a function of the inserted yield points (i.e., it is set independent of the status of a yield point). This is different than Dean1 which implements a selector mechanism

incorporating a trigger that is a function of received events, transactions or state depending upon a particular functional block that is sampled (See Figure 2 of Dean1). Upon receipt of a particular combination of received event, transactions or state, the trigger (in Dean1) will either enable or disable a counter counting operation which determines a sampling rate. Thus, Dean1 describes a counter based approach for sampling. Again, it is emphasized that this is different than in the present invention which is a software-based approach that performs active checking independent of the identification of a yield point, and is not counter based as in Dean1.

With respect to the Examiner's rejection of Claims 4, 5 and 28-29, the Examiner cites Brown as teaching the use of implementing a variable for tracking sampling operations. Applicants respectfully submit that Brown is unrelated to the teachings of the present invention as it is concerned with the implementation of a zener diode to provide improved random number generators that meet a specified degree of randomness. The task of generating a random number involves counting the number of zero's and one's in a generated pulse a fixed number of times. The software-based sampling approach taken in the present invention however, is not dependent on a valid randomness decision. In fact, in the counter-based approach embodiment of the invention, the order of the yield points and the samples taken may be deterministic. The approach of the present invention is not interested in adjusting control voltage (zener diode biasing point), nor has anything to due with voltage. As such, applicants fail to see the relevance of this reference in the context of the present invention.

In response to the Examiner's rejection of Claim 11 (dependent upon Claim 3), Claim 22 (dependent upon Claim 17) and Claim 33 (dependent upon Claim 27), the Examiner

alleges that McDevitt's call back service is related to a call stack runtime data structure. Applicants respectfully submit that a call back is a function call from a service back to the client making the request, whereas a call stack runtime data structure is a representation as a stack data structure of the function calls to the current function where each entry in the stack represents a function that has not finished executing. Thus, respectfully, the Examiner's reliance upon McDevitt is misplaced and the Examiner is respectfully requested to withdraw the rejection of these claims under 35 U.S.C. §103(a).

With respect to the other references cited by the Examiner but not applied against any of the claims, applicants respectfully submit the following:

Ungar's patent U.S. 6,282,702 entitled "Method and Apparatus of Translating and Executing Native Code in a Virtual Machine Environment" discusses yield points as a mechanism to support thread switching. Although the present invention utilizes yield points, it is not for this purpose; rather, in the invention, yield points are used as a mechanism to sample the state of the runtime environment.

The Cocchi et al. patent U.S. 6,442,751 entitled "Determination of Local Variable Type and Precision in the Presence of Subroutines", it is respectfully submitted, is not related to the present invention. In particular, this reference is concerned with generating precise type information for slots in the stack frame for local variables at a particular point in a subroutine. This is an issue because a slot can be reused for different local variables, and hence can have different types at different points in the subroutine. The precise type information is determined by analyzing the byte codes of the method. JSR (jump to subroutine) code complicates the computation of precise type information. Precise type information is required for GC and debugging purposes. One could consider the "particular

point in the subroutine" as a potential candidate as a yield point. Nevertheless, they are using a particular point in the subroutine for GC and debugging, but not for sampling the runtime state, (i.e. call stack).

This application is now believed to be in condition for allowance, and a Notice of Allowance is respectfully requested. If the Examiner believes a telephone conference might expedite prosecution of this case, it is respectfully requested that the Examiner call applicant's attorney at (516) 742-4343.

Respectfully submitted,

Steven Fischman
Registration No.: 34,594

Scully, Scott, Murphy & Presser
400 Garden City Plaza
Garden City, New York 11530
(516) 742-4343
SF:gc